



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2003-0034534
Application Number

출원 년 월 일 : 2003년 05월 29일
Date of Application MAY 29, 2003

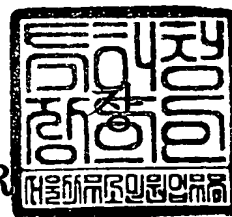
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 11 월 18 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.05.29
【발명의 명칭】	망관리 시스템의 운용자 접근 제어에 대한 보안 방법
【발명의 영문명칭】	Security method for access control of Network Management System
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	박상수
【대리인코드】	9-1998-000642-5
【포괄위임등록번호】	2000-054081-9
【발명자】	
【성명의 국문표기】	이현숙
【성명의 영문표기】	LEE, HYUN SOOK
【주민등록번호】	700626-2110829
【우편번호】	449-753
【주소】	경기도 용인시 수지읍 동성1차아파트 108동 1305호
【국적】	KR
【우선권주장】	
【출원국명】	KR
【출원종류】	특허
【출원번호】	10-2003-0010509
【출원일자】	2003.02.19
【증명서류】	미첨부
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 박상수 (인)

1020030034534

출력 일자: 2003/11/25

【수수료】

【기본출원료】	20	면	29,000	원
---------	----	---	--------	---

【가산출원료】	6	면	6,000	원
---------	---	---	-------	---

【우선권주장료】	1	건	26,000	원
----------	---	---	--------	---

【심사청구료】	6	항	301,000	원
---------	---	---	---------	---

【합계】	362,000	원		
------	---------	---	--	--

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 현장에서 사용하고 있는 시스템 어플리케이션 프로토콜의 버전을 바꾸지 않고도 접근 제어(access control)를 할 수 있는 방법을 제공하기 위한 것으로서, 운용자는 사용자 인증을 위해 아이디(ID)와 패스워드(password)를 입력하고, 사용자 인증과정이 성공되면, TCP/IP 또는 UDP/IP를 통해 관리하고자 하는 시스템의 어플리케이션 계층에 접근하게 되는데, 이때 운용자가 사용하는 단말기의 IP 주소가 미리 설정된 IP 주소인지를 확인하기 위해, 보안 모듈을 통해 어플리케이션 계층에 접근하도록 구성됨으로써, 보안 기능이 없는 망관리 인터페이스 버전을 운용하는 망의 경우 버전 업그레이드 과정없이 간단하게 보안 모듈을 추가함으로써 시스템의 보안 문제를 해결해준다.

【대표도】

도 4

【색인어】

SNMP, CLI, TL1, 프로토콜 보안, OAMP

【명세서】

【발명의 명칭】

망관리 시스템의 운용자 접근 제어에 대한 보안 방법{Security method for access control of Network Management System}

【도면의 간단한 설명】

도 1은 본 발명에 적용되는 간이 망 관리 프로토콜(SNMP) 및 CLI(TL1)을 이용한 망관리 시스템을 설명하기 위한 블록도,

도 2는 종래 기술에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면,

도 3은 본 발명에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면,

도 4는 본 발명에서 정의된 MIB을 이용하여 구성되는 필터링 테이블의 실예도,

도 5는 본 발명에 따른 망관리 시스템의 운용자 접근 제한 보안 과정에 대한 흐름도.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<6> 본 발명은 시스템 어플리케이션 프로토콜의 버전을 바꾸지 않고도 접근제어를 할 수 있도록 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법에 관한 것이다.

<7> 현재, 인터넷을 포함한 네트워크 관련 장비들은 대부분 네트워크를 관리하고, 각 네트워크 장치들의 동작을 감시하기 위해 간이 망 관리 프로토콜 (Simple Network Management Protocol: SNMP)을 기반으로 하는 네트워크 관리 프로토콜을 이용한다. 이러한 SNMP는 가장 일반적인 네트워크 관리 프로토콜이라 할 수 있으며, 각각 V1/V2/V3의 버전으로 발전되면서 그

기능이 많이 향상되었다. 대부분의 네트워크 시스템은 이러한 SNMP를 이용하는 그래픽 운용자 인터페이스 (Graphic User Interface: GUI) 기반의 엘리먼트 관리 시스템(Element Management System:EMS)과 외부 터미널을 통하여 명령어를 직접 수신하여 처리하는 명령어 인터페이스 (Command Line Interface :CLI)를 함께 제공한다.

<8> 도 1은 본 발명에 적용되는 망 관리 프로토콜(SNMP) 및 CLI(TL1)을 이용한 망관리 시스템을 설명하기 위한 블록도이다. 도 1을 참조하면, 시스템(100)에서 제공하는 망 관리 인터페이스(management interface)에는 "TL1/CLI(Transaction Language 1/Command Line Interface)(110)"과 "SNMP 대리인(120)"이 있다. 시스템은 이러한 관리 채널(management channel)들을 통해 시스템의 구성, 정보, 성능 등을 관리하게 된다.

<9> TL1(110)의 경우, 외부의 콘솔(200)들과 시리얼 포트(serial port)로 직접 연결하여 시스템(100)을 관리할 수도 있고, 공중망(public network)(300)을 통한 텔넷(telnet)(400)으로 원격 제어(remote control)도 가능하다.

<10> 한편, SNMP 대리인(120)은 UDP(User Datagram Protocol)/IP를 이용하여 공중망(300)을 통해 주로 EMS(Element Management System) 서버(500)와 연결되어 사용되어지나, OSI(Open Systems Interconnection) CLNP(Connectionless Network Protocol) 위의 프로토콜로 사용되어 지기도 한다.

<11> 이와 같은 TL1(110)과 SNMP 대리인(120)은 각각 OAMP(Operations Administration Maintenance Provisioning)(130)로부터 IPC(Interprocess Communication)를 통해 필요한 데이터를 가지고 오기도 하고, 변경시키기도 한다.

- <12> 도 2는 종래 기술에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면이다. 도 2를 참조하면, 텔넷 터미널(400) 또는 EMS 서버(500)는 물리 계층을 통해 데이터링크 계층과 연결되어, TCP/IP 또는 UDP/IP 방식으로 어플리케이션 계층{SNMP/ 텔넷/ TFTP(Trivial File Transfer Protocol)}에 접근하게 된다.
- <13> 상기와 같이 구성되는 망관리 시스템에 사용되는 SNMP는 V1/ V2/ V3까지 사용되고 있는데, 그 중 SNMP V1/ V2의 접근 제한 방법으로는 "read-only"/"read-write" 커뮤니티 (community)를 체크하는 방법이 주로 사용되고 있으며, SNMP V3는 보안 모듈이 프로토콜 내에 존재하고 있다.
- <14> 커뮤니티는 매니저와 대리인간에 정의된 패스워드 방식의 내역을 의미한다.
- <15> 예컨대, SNMP V1/ V2의 일반적인 커뮤니티는 "read-only"의 경우 "public", "read-write"의 경우 "private"로 많이 사용되고 있다. 또한, 어떤 시스템들은 이러한 커뮤니티들이 하드 코딩(hard-coding)되어 있어, 커뮤니티 변경이 용이하지 않다. 이와 같은 시스템들에 있어서는 커뮤니티 패스워드 자체가 노출되어 있어서 운용자 이외의 다른 사람이 임의의 장소에서 망 관리 시스템에 접근할 수 있게 되므로 보안상 문제가 발생할 수 있다.
- 【발명이 이루고자 하는 기술적 과제】**
- <16> 본 발명은, 상기와 같은 문제점을 해결하기 위하여 제안된 것으로, 현장에서 사용하고 있는 시스템 어플리케이션 프로토콜의 버전을 바꾸지 않고도 접근 제어(access control)를 할 수 있는 방법을 제공하는데 그 목적이 있다.

【발명의 구성 및 작용】

- <17> 상기의 목적을 달성하기 위한 본 발명에 따른 망관리 시스템의 운용자 접근 제어에 대한 보안 방법은, 외부의 운용자가 TCP/IP(Transmission Control Protocol/Internet protocol) 또는 UDP/IP(User Datagram Protocol/Internet protocol)를 통해 IP(Internet protocol) 주소가 미리 설정되어 있는 IP 주소인지를 확인할 수 있도록 IP 필터링을 수행하는 제 1 단계; 및 상기 제 1 단계에 의해 통신 시스템에 접속시, 아이디(ID)/패스워드(Password) 입력 또는 커뮤니티(community)들을 설정하여 통신 시스템에 접속하는 제 2 단계를 포함하여 이루어지는 것을 특징으로 한다.
- <18> 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일실시예를 상세히 설명한다.
- <19> 본 발명에 적용되는 간이 망 관리 프로토콜(SNMP) 및 CLI(TL1)을 이용한 망관리 시스템의 구성은 종래기술과 동일하므로 그 설명은 생략한다.
- <20> 도 3은 본 발명에 따른 망관리 시스템을 OSI 참조 모델과 관련하여 설명한 도면이다.
- <21> 만약 TL1(110)을 이용하여 망관리 동작을 수행하는 경우, 도 1 및 도 3을 참조하면, 먼저 운용자는 사용자 인증을 위해 아이디(ID)와 패스워드(password)를 입력한다. 사용자 인증과정이 성공되면, TCP/IP 또는 UDP/IP를 통해 관리하고자 하는 시스템의 어플리케이션 계층에 접근한다. 이때 운용자가 사용하는 단말기의 IP 주소가 미리 설정된 IP 주소인지를 확인하기 위해, 보안 모듈(150)을 통해 어플리케이션 계층에 접근하도록 구성된다.
- <22> 즉, IP 망(IP network)(예컨데, 도 1에서의 공중망)을 통한 원격 관리 채널(remote management channel)인 텔넷 터미널(400)은 아이디/패스워드 보안 장치에다 텔넷 프로토콜

(telnet protocol)을 사용하는 운용 단말기(terminal)의 IP 주소(IP address)가 보안의 열쇠가 될 수 있는 필터링(filtering) 기능을 가진다.

- <23> 여기서, 이 모듈은 "TL1" 기능을 구현한 "CLI(Command Line Interface)" 작업(task)과는 전혀 별개의 것으로 구현이 되어 있다.
- <24> 그리고, SNMP V1/ V2는 기본적인 보안이 커뮤니티를 통해 이루어지는데, 이는 "read-only" 커뮤니티와 "read-write" 커뮤니티로 이루어지며 주로 변경이 가능하지 않는 경우가 많다.
- <25> 본 발명에서는 이러한 커뮤니티의 보안을 위해, 각 커뮤니티는 "TL1" 명령어로만 변경이 가능하도록 한다. 즉, "SNMP"를 이용해서는 커뮤니티를 읽어볼 수도 없고, 변경도 불가능하므로 항상 운용자가 "TL1" 명령어를 통해 알고 있어야만 EMS 서버(500)와의 통신이 가능하다. 또한, 커뮤니티의 변경시에도 관리하는 EMS 서버(500)와의 협의가 요구되어진다.
- <26> 그리고, SNMP V1/ V2가 UDP/IP나 TCP/IP를 사용할 경우 "TL1"과 마찬가지로 IP 필터링을 통해 운용자의 IP 주소를 키(key)로 하여 보안이 이루어지며 이것을 MIB으로 표현해 놓은 것이 표 1 내지 표 17이 된다.
- <27> 표 1은 입력 패킷(ingress packets)을 필터링하기 위한 시스템의 정책 아이디를 표시한다. 이 객체의 값은 "entFilterPolicyTable"의 "entFilterPolicyId"의 값이다.
- <28> 그리고, 'DEFVAL'는 모든 입력 패킷들을 인정한다.
- <29>

【표 1】

```

entIngressFilterPolicyId OBJECT-TYPE
    SYNTAX      INTEGER (0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
            Indiactes the policy id of system for filtering ingress packets.
            The value of this object is that of entFilterPolicyId
            in entFilterPolicyTable.

            'DEFVAL' : accept all ingress packets
        "
    DEFVAL { 0 }
    ::= { entConfig 13 }

```

<30> 그리고, 표 2는 출력 패킷(egress packets)을 필터링하기 위한 시스템의 정책 아이디를 표시한다. 이 객체의 값은 "entFilterPolicyTable"의 "entFilterPolicyId"의 값이다.

<31> 그리고, 'DEFVAL'는 모든 출력 패킷들을 버리지 않는다.

<32> 【표 2】

```

entEgressFilterPolicyId OBJECT-TYPE
    SYNTAX      INTEGER (0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "
            Indiactes the policy id of system for filtering egress packets.
            The value of this object is that of entFilterPolicyId
            in entFilterPolicyTable.

            'DEFVAL' : not discard all egress packets
        "
    DEFVAL { 0 }
    ::= { entConfig 14 }

```

- <33> 그리고, 표 3은 입/출력 패킷에 대한 시스템의 필터링 정책을 담고 있다. 이 표에서 "row"는 "entFilterIpTable"과 같은 프로토콜 표에서의 "row"를 가리킨다.
- <34> 이 표에서 "row"를 생성하는데 있어서, "entFilterPolicyPointer" 객체에 의해 가리켜지는 "row"가 처음으로 생성된다.
- <35> 그리고, 이 표에서 "row"를 삭제하는데 있어서, "entFilterPolicyPointer" 객체에 의해 가리켜지는 "row"가 처음으로 삭제된다.
- <36> **【표 3】**

```
entFilterPolicyTable OBJECT-TYPE
    SYNTAX SEQUENCE OF EntFilterPolicyEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "
            This table contains the filtering policies of system
            on ingress/egress packet.
            A row in this table is pointing a row in protocol table
            such as entFilterIpTable.
            For creating a row in this table, the row that is pointed
            by entFilterPolicyPointer object was first created.
            And for destroying a row in this table, the row that is pointed
            by entFilterPolicyPointer object was first destroyed.
        "
    ::= { entConfig 15 }
```

- <37> 그리고, 표 4에서 각각의 엔트리는 시스템의 필터링 정책을 대표하는 파라미터들의 리스트를 구성한다.
- <38>

【표 4】

```
entFilterPolicyEntry OBJECT-TYPE
    SYNTAX      EntFilterPolicyEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "
            Each entry consists of a list of parameters that
            represents filtering policy on a system.
        "
    INDEX        { entFilterPolicyIndex }
    ::= { entFilterPolicyTable 1 }

EntFilterPolicyEntry ::= SEQUENCE {
    entFilterPolicyIndex      INTEGER,
    entFilterPolicyId         INTEGER,
    entFilterPolicyPointer    RowPointer,
    entFilterPolicyRowStatus  RowStatus
}
```

<39> 그리고, 표 5는 "entFilterPolicyTable"로의 인덱스이다.

<40> 【표 5】

```
entFilterPolicyIndex OBJECT-TYPE
    SYNTAX      INTEGER(1..9)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "
            The index into the entFilterPolicyTable.
        "
    ::= { entFilterPolicyEntry 1 }
```

<41> 그리고, 표 6은 입력 또는 출력 정책의 식별을 나타낸다. 같은 정책 아이디는 이 표에서 많은 "row"들에 속할 수 있다.

<42>

【표 6】

entFilterPolicyId OBJECT-TYPE	
SYNTAX	INTEGER(1..255)
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Indicates the identification of ingress or egress policy.
	A same policy id could belong to many rows in this table.
	"
::= { entFilterPolicyEntry 2 }	

<43> 그리고, 표 7은 "entFilterIp"와 같은 프로토콜 표에서 "row"에 대한 포인터를 대표한다.
그 값은 프로토콜 표에서 첫번째 세로열 객체에 대한 이름이다.

<44> 예를 들면, 이 객체의 실예의 값인 "entFilterIpIndex.3"은 "entFilterIp"표의 세번째 "row"를 가리킨다.

<45> 【표 7】

entFilterPolicyPointer OBJECT-TYPE	
SYNTAX	RowPointer
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Represents a pointer to a row in protocol table such as
	entFilterIp table. The value is the name of the instance of the first columnar object in
	the protocol table.
	For example, entFilterIpIndex.3 that is the value of the instance
of	this object would point to the 3rd row in the entFilterIp table.
	"
::= { entFilterPolicyEntry 3 }	

<46> 그리고, 표 8에서 객체는 새로운 "row"를 생성하거나, 수정하거나, 이 표에서 존재하는 "row"를 삭제하는데 이용된다.

<47> 만약, "entFilterIp"표와 같은 프로토콜 표의 관계된 "row"가 생성되지 않는다면, 이 표에서 "row"는 생성되어질 수 없을 것이다.

<48> 【표 8】

```
entFilterPolicyRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "
            This object is used to create a new row or modify or
            delete an existing row in this table.

            If the related row of protocol table such as entFilterIp table
            wasn't created, a row in this table could have not been created.

            The related row of protocol table should have been first
            destroyed before a row in this table is destroyed.
        "
    ::= { entFilterPolicyEntry 4 }
```

<49> 그리고, 표 9는 IP 프로토콜 상의 필터 정책에 대한 세부 사항을 담고 있다.

<50> 【표 9】

```
entFilterIpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EntFilterIpEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "
            This table contains the details of a filter policy over IP protocol.
        "
    ::= { entConfig 16 }
```

<51> 그리고, 표 10에서 각각의 엔트리는 IP 프로토콜 상의 필터 정책을 대표하는 파라미터들의 리스트를 구성한다.

<52> 【표 10】

```

entFilterIpEntry OBJECT-TYPE
    SYNTAX      EntFilterIpEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "
            Each entry consists of a list of parameters that
            represents a filter policy over IP protocol .
        "
    INDEX        { entFilterIpIndex }
    ::= { entFilterIpTable 1 }
EntFilterIpEntry ::= SEQUENCE {
    entFilterIpIndex      INTEGER,
    entFilterIp           IpAddress,
    entFilterIpMask       IpAddress,
    entFilterIpPortNum    INTEGER,
    entFilterIpProtocol   INTEGER,
    entFilterIpControl    INTEGER,
    entFilterIpRowStatus  RowStatus
}

```

<53> 그리고, 표 11은 "entFilterIpTable"로의 인덱스이다.

<54> 【표 11】

```

entFilterIpIndex OBJECT-TYPE
    SYNTAX      INTEGER(1..9)
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "
            The index into the entFilterIpTable.
        "
    ::= { entFilterIpEntry 1 }

```

<55> 그리고, 표 12는 필터 정책에 적용된 IP 주소를 지시한다.

<56>

【표 12】

entFilterIp	OBJECT-TYPE
SYNTAX	IpAddress
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Indicates ip address applied a filter policy.
	"
DEFVAL	{ '00000000'h }
::=	{ entFilterIpEntry 2 }

<57> 그리고, 표 13은 IP 주소의 마스크를 지시한다. 그리고, "entFilterIpProtocol"이 텔넷 (telnet)일 때, 시스템은 항상 'DEFVAL'을 이 객체의 실시예에 적용한다.

<58> 【표 13】

entFilterIpMask	OBJECT-TYPE
SYNTAX	IpAddress
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Indicates the mask of ip address.
	When entFilterIpProtocol is telnet,
	system always applies 'DEFVAL' to the instance of this object.
	"
DEFVAL	{ 'ffffffff'h }
::=	{ entFilterIpEntry 3 }

<59> 그리고, 표 14는 필터 정책에 적용되는 포트 넘버를 지시한다.

<60>

【표 14】

entFilterIpPortNum OBJECT-TYPE	
SYNTAX	INTEGER
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Indicates the applied port number to a filter policy.
	"
::= { entFilterIpEntry 4 }	

<61> 그리고, 표 15는 필터 정책에 적용되는 프로토콜을 지시한다.

<62> 【표 15】

entFilterIpProtocol OBJECT-TYPE	
SYNTAX	INTEGER { snmp(1), telnet(2), tftp(3) }
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Indicates the applied protocol over IP protocol to a filter
	policy.
	"

<63> 그리고, 표 16은 패킷을 받아들일지 버릴지를 결정한다.

<64> 【표 16】

entFilterIpControl OBJECT-TYPE	
SYNTAX	INTEGER { discard(1), accept(2) }
MAX-ACCESS	read-create
STATUS	current
DESCRIPTION	"
	Determines whether to discard or accept a packet.
	"
::= { entFilterIpEntry 6 }	

<65> 그리고, 표 17에서 이 객체는 새로운 "row"를 생성하거나, 수정하거나, 이 표에서 존재하는 "row"를 삭제하는데 이용된다.

<66> 【표 17】

```
entFilterIpRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "
            This object is used to create a new row or modify or
            delete an existing row in this table.
        "
    ::= { entFilterIpEntry 7 }
```

<67> 다음은 필터링에 관한 동작을 표 1 내지 표 17에 표현되어 있는 MIB 객체로 설명하면 다음과 같다. 먼저 "entFilterIpTable"에 있는 객체들을 필터링시키고자하는 범위로 셋팅(setting)한 후 "row"를 생성(create)시킨다. 이때, "entFilterIpProtocol"의 의미는 "IP 상의 프로토콜(protocol over ip)"라고도 표현되어질 수 있다.

<68> 여기서, 필터링하고자 하는 프로토콜은 SNMP/ 텔넷/ TFTP(Trivial File Transfer Protocol) 등이 가능하다. 그리고, "entFilterIpControl"에서는 패킷을 버리거나(discard) 받아들이는(accept) 것이 설정가능한 값이다.

<69> 해당 "row"가 출력 정책(egress policy)으로 사용되어질 경우에, SNMP 패킷은 요청(request)은 받아들여지나 응답 패킷(response packet)이 내보내지지 않는다. 물론 트랩에도 마찬가지로 적용되어져 트랩 패킷도 등록된 EMS 서버(500)로 내보내지지 않게 된다. 반면, 해당 "row"가 입력 정책(ingress policy)으로 사용되어질 경우에는 반대로 동작하게 된다. 일단 "entFilterIpTable"의 "row"가 생성되고 난 후에는 "entFilterPolicyTable"의 "row"가 생성되어

야 한다. 이 표(table)의 목적은 하나의 정책에 여러 가지 "row"들이 포함될 수도 있는 다양성을 위해 구현된다.

- <70> 그리고, "entFilterPolicyPointer"는 앞서 만들어진 "entFilterIpTable"의 "row"를 포인팅(pointing)하고 있다. 이 때 "entFilterPolicyId"는 같은 값을 여러개의 "row"들이 가지는 것도 가능한 구조로 구현된다. 그리고, "entIngressFilterPolicyId"와 "entEgressFilterPolicyId"의 값을 셋팅한다. 이 값들은 시스템과 타 장비간의 주고받는 패킷 전반에 영향을 미치게 된다.
- <71> 상기의 표 1 ~ 표 17에 의해 표현되는 객체들에 대하여 실예를 들어 설명하면 다음과 같다.
- <72> 도 4는 본 발명에서 정의된 MIB을 이용하여 구성되는 필터링 테이블의 실예도이다. 도 4를 참조하면, 운용자가 선택한 PolicyId(PID)번호에 대한 필드와, 각 PolicyId에 대응하는 포인터값을 갖는 포인터 필드와, 해당 "row"의 상태를 나타내는 Row Status 필드로 이루어지는 FilterPolicy 테이블(T1); FilterPolicy 테이블(T1)의 포인터값을 인덱스 번호로 하는 인덱스 필드와, 해당 "row"별로 IP 주소를 나타내는 IP 필드와, IP 주소를 마스킹하여 그룹을 설정할 수 있도록 하는 Mask 필드와, 포트번호 필드와, 프로토콜 필드와, 컨트롤 필드와, Row Status 필드로 이루어지는 FilterIp 테이블(T2)로 이루어진다.
- <73> 상기 FilterPolicy 테이블(T1)의 PolicyID 필드, Pointer 필드 그리고 Row Status 필드는 Integer(정수)형태이다. 단, PolicyID 필드와 Pointer 필드의 정수는 숫자 자체를 의미하지만, Row Status 필드의 정수는 그 숫자가 나타내는 의미가 있다.

- <74> 예컨데, 1이면 "row"의 상태가 active이고, 2이면 서비스되지 않는 상태(notInService), 3이면 준비되지 않음(notReady), 4이면 row 생성(createAndGo), 5이면 row 생성중(createAndWait), 6이면 row 삭제(destroy)를 의미하도록 정의되어 있다.
- <75> 한편, FilterIp 테이블(T2)의 경우는, 인덱스 번호 필드, 포트번호 필드, 프로토콜 필드, 컨트롤 필드, Row Status 필드는 정수 형태이며, IP 주소 필드와, IP 주소 마스크 필드는 IP 주소 형식(xxx.xxx.xxx.xxx)이다. 단, 프로토콜 필드와 컨트롤 필드 그리고 Row Status 필드의 정수는 그 숫자가 나타내는 의미가 있다.
- <76> 예컨데, 프로토콜 필드가 1이면 프로토콜 종류가 SNMP임을 의미하고, 2이면 Telnet을 의미하고, 3이면 TFTP를 의미하도록 정의되어 있다.
- <77> 또한, 컨트롤 필드가 1이면 discard를 의미하고, 2이면 accept를 의미하도록 정의되어 있다.
- <78> 그리고, Row Status 필드는 FilterPolicy 테이블(T1)의 Row Status 필드와 동일한 방식으로 숫자가 정의된다.
- <79> 이하, 상기의 테이블을 이용하여 실제로 운용자가 접근 허용/거부를 수행하는 과정을 살펴보기로 한다.
- <80> 도 5는 본 발명에 따른 망관리 시스템의 운용자 접근 제한 보안 과정에 대한 흐름도이다. 도 5를 참조하면, 먼저 패킷 처리방법을 어떻게 할 것인지에 대한 정책을 결정하여 결정된 정책에 대한 정책 Id(PolicyId:PIId)를 결정한다(S10).
- <81> S10에서 결정된 PIId 값에 따라 해당 값을 갖는 "row"를 FilterPolicy 테이블(T1) 상에서 찾는다(S20).

- <82> S20에서 찾아진 "row"의 Pointer 값을 읽어(S30), 포인터값을 인덱스 번호로 하는 FilterIp 테이블(T2)에서 해당 "row"를 찾아 해당 "row"에 설정된 조건들(IP 주소, 마스크, 포트번호, 프로토콜, IP 컨트롤 방법)에 따라 패킷을 처리한다(S40).
- <83> 예를 들어, 정책 Id(PId)가 100으로 결정되면, FilterPolicy 테이블(T1)의 인덱스 번호 1번에 해당하는 "row"가 될 것이다. 인덱스 번호 1에 해당하는 "row"의 포인터값은 "1"이므로 FilterIp 테이블(T2)의 인덱스 번호 1에 해당하는 "row"에 해당하는 조건들을 수행하게 될 것이다.
- <84> 따라서, 정책 Id가 100으로 결정된 상황인 경우 FilterIP 테이블의 1번 "row"에서 설정된 IP 주소와 다른 IP 주소의 단말기로부터 운용자 접근이 시도된다면, 접근이 실패할게 될 것이다. 또한 IP 주소가 같더라도 포트번호가 미리 설정된 번호(161)와 다른 번호로 패킷이 송수신 되어도 운용자 접근이 실패될 것이다.
- <85> 다음으로 SNMP V1/ V2를 위한 커뮤니티 변경 및 조회에 관한 "TL1" 명령어의 수행 결과에 대한 실예를 표 18과 같이 제시한다.
- <86> 【표 18】

```

SU-WON> rtrv-community;

IP C01240
<

SU-WON 2002-02-02 01:56:40
M C01240 COMPLD
"RD=SamsungAcemap,WR=K_SAMSUNG_Acemap2000_set,TR=SS_Acemap_Trap"
/* RTRV-COMMUNITY; [C01240] */
;

```

- <87> 여기서, "RD"는 "read-only" 커뮤니티, "WR"은 "read-write" 커뮤니티, "TR"은 "트랩(trap)" 커뮤니티를 의미한다. 이것은 "TL1" 명령어로만 변경 및 조화가 가능하고, 변경 시 EMS 서버(500)에서의 관리를 위해서는 EMS 서버(500)에서도 커뮤니티의 변경이 되어야만 한다.
- <88> 상기와 같이 각 커뮤니티 패스워드를 변경하면, 일반적인 패스워드와 다르므로 일반인들에게 쉽게 노출되지 않게 된다.
- <89> 이상 본 발명의 바람직한 실시예에 대해 상세히 기술되었지만, 본 발명이 속하는 기술분야에 있어서 통상의 지식을 가진 사람이라면, 첨부된 청구 범위에 정의된 본 발명의 기술적 사상 및 범위를 벗어나지 않으면서 본 발명을 여러 가지로 변형 또는 변경하여 실시할 수 있음을 알 수 있을 것이다. 따라서 본 발명의 앞으로의 실시예들의 변경은 본 발명의 기술을 벗어날 수 없을 것이다.

【발명의 효과】

- <90> 이상 설명한 바와 같이, 본 발명에 따르면, 망관리 프로토콜 버전이 EMS와 동일 버전으로 운용 되고 있는 시스템에서, SNMP V1/ V2을 보안 기능이 제공되는 SNMP V3로 업그레이드하지 않고도 IP 필터링을 수행하는 보안 모듈을 추가함으로써 간단하게 망관리 인터페이스에 접속시 보안을 유지할 수 있게 된다.

【특허청구범위】**【청구항 1】**

외부의 운용자가 TCP/IP(Transmission Control Protocol/Internet protocol) 또는 UDP/IP(User Datagram Protocol/Internet protocol)를 통해 IP(Internet protocol) 주소가 미리 설정되어 있는 IP 주소인지를 확인할 수 있도록 IP 필터링을 수행하는 제 1 단계; 및

상기 제 1 단계에 의해 통신 시스템에 접속시, 아이디(ID)/패스워드(Password) 입력 또는 커뮤니티(community)들을 설정하여 통신 시스템에 접속하는 제 2 단계를 포함하여 이루어지는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 2】

제 1 항에 있어서, 상기 제 1 단계는,

MIB(Management Information Base) 으로 구현된 객체들을 필터링시키고자하는 범위로 셋팅(setting)한 후 "row"를 생성(create)시키는 제 11 단계;

입력되거나 출력하려는 SNMP 패킷에 대해 버리거나(discard) 받아들이는(accept) 것이 설정가능하도록 하는 제 12 단계;

상기 "row"가 출력 정책(egress policy)으로 사용되어질 경우에, 상기 SNMP 패킷은 요청(request)은 받아들여지나 응답 패킷(response packet)은 출력하지 않도록 하는 제 13 단계; 및

상기 "row"가 입력 정책(ingress policy)으로 사용되어질 경우에, 상기 SNMP 패킷은 요청(request)은 받아들여지지 않으나 응답 패킷(response packet)은 출력하도록 하는 제 14 단계를 포함하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 3】

제 2항에 있어서, 제 11단계는,

패킷 처리방법을 어떻게 할 것인지에 대한 정책 Id(PolicyId:PIId)를 결정하는 제 110단계;

제 110단계에서 결정된 PIId 값에 따라 해당 값을 갖는 "row"를 FilterPolicy 테이블 상에서 찾는 제 120단계:

제 120단계에서 찾아진 "row"의 Pointer 값을 읽는 제 130단계: 및

제 130단계에서 읽은 포인터값을 인덱스 번호로 하는 FilterIp 테이블에서 해당 "row"를 찾아 해당 "row"에 설정된 IP 주소 및 포트번호에 대한 조건에 따라 운용자 접근 허용 여부를 결정하여 패킷을 처리하는 제 140단계로 이루어지는 것을 특징으로 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 4】

제 3항에 있어서, 제 140단계의 운용자 접근 허용 여부를 결정하기 위한 조건 항목들이 기록된 FilterIp 테이블은,

정책 Id에 대응하는 포인터값을 인덱스로 하는 인덱스 번호 필드와, IP 주소 필드와, IP 주소 마스크 필드와, 포트번호 필드와, 프로토콜 필드와, 컨트롤 필드와, Row Status 필드로 이루어지는 것을 특징으로 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 5】

제 4항에 있어서,

인덱스 번호 필드, 포트번호 필드, 프로토콜 필드, 컨트롤 필드, Row Status 필드의 syntax(구문)는 Integer(정수)이고,

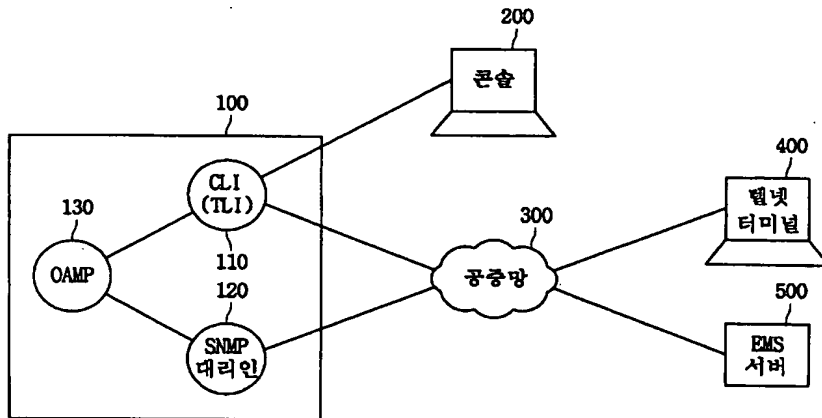
IP 주소 필드와, IP 주소 마스크 필드의 syntax는 IP 주소 형식인 것을 특징으로 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【청구항 6】

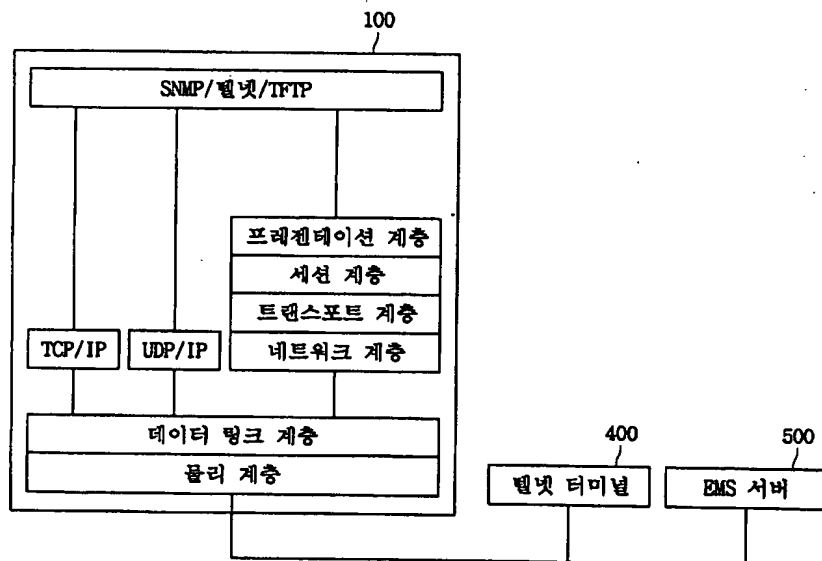
제 1 항에 있어서, 상기 외부의 운용자는 텔넷 터미널 또는 EMS 서버인 것을 특징으로 하는 망관리 시스템의 운용자 접근 제어에 대한 보안 방법.

【도면】

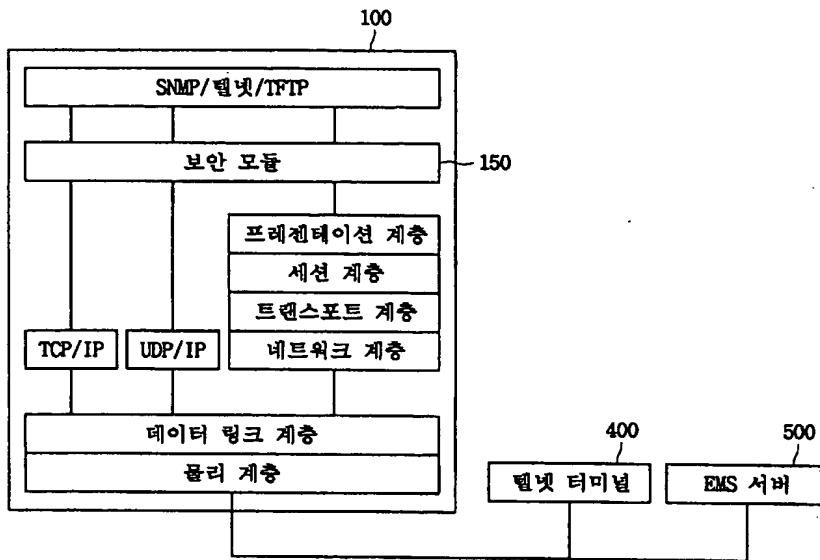
【도 1】



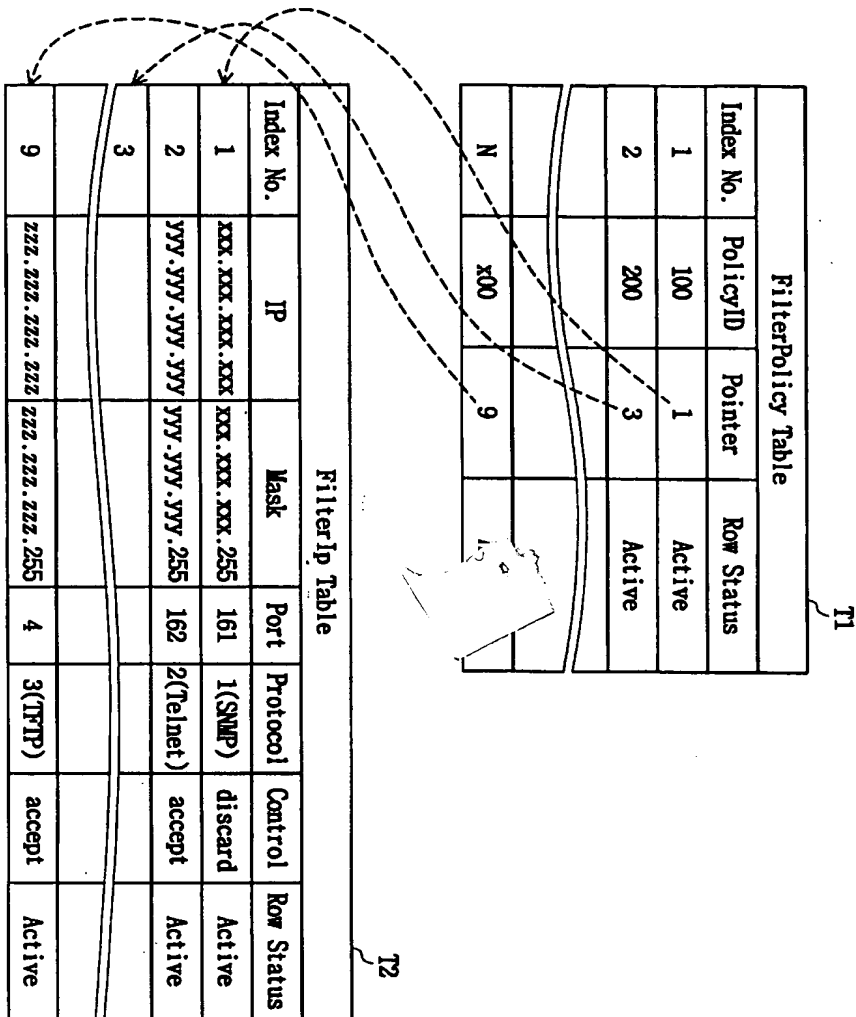
【도 2】



【도 3】



【도 4】



【도 5】

